

Revista Bibliotecas
Vol. XXIX, No. 1
ene.-jun., 2011
pp.

El *software* libre: Una alternativa para automatizar unidades de información¹

Free software: an alternative to automate information units

Máster Ricardo Chinchilla Arley²

Resumen

El documento describe la problemática presupuestaria en las unidades de información y los altos costos del software comercial especializado en la automatización de bibliotecas. Describe los orígenes del *software* libre y su significado. Menciona los tres niveles de automatización dentro de una biblioteca: automatización de catálogos, generación de repositorios y automatización integral. Señala las distintas aplicaciones de *software* libre para cada uno de los niveles y ofrece una serie de ventajas y desventajas en la utilización de este tipo de productos. Se concluye que el proyecto de automatización es arduo pero lleno de satisfacciones, haciendo énfasis en que no existe un proyecto libre de costos, ya que si bien es cierto que el *software* libre es gratuito, existen otros costos relacionados con la implementación, capacitación y puesta en marcha del proyecto.

Palabras clave:

Automatización de bibliotecas, *Software* libre, Bibliotecas automatizadas

Abstract

The document begins by describing the problem of budget information units and the high cost of commercial software that specializes in library automation. Describes the origins of free software and its meaning. Mentioned the three levels of automation in library: catalog automation, generation of repositories and full automation. Mentioned the various free software applications for each of the levels and offers a number of advantages and disadvantages in the use of these products. Concludes that the automation project is hard but full of satisfaction, emphasizing that there is no cost-free project, because if it is true that free

¹ Recibido el 7 de enero de 2011, aprobado el 10 de marzo de 2011

² Profesor, Escuela de Bibliotecología y Ciencias de la Información. Universidad de Costa Rica.

software is free, there are other costs related to implementation, training and commissioning project progress.

Keywords

Library automation, Free software, Automated libraries

1. Introducción

Mucho se habla hoy en día acerca de las tecnologías de información y las telecomunicaciones (TIC) y su importancia en el desarrollo de un país. De hecho, existen numerosos índices de desarrollo que utilizan las TIC como una variable determinante para definir la posición de un país o región dentro de su *ranking*.

Sin embargo, paradójicamente, las unidades de información (bibliotecas, centro de documentación, centros referenciales, etc.), a pesar de ser los proveedores del componente estratégico y la materia prima para el desarrollo de la Sociedad de la Información, se encuentran en el último lugar en las prioridades de inversión en las organizaciones. Por otro lado, el costo del *software* comercial para la automatización integral de unidades de información (UI) continúa siendo sumamente elevado y sus requerimientos de equipo de cómputo implican grandes inversiones, lo cual desestimula la generación de proyectos de automatización. Generalmente, los grandes proyectos de inversión en el campo son realizados solamente por las universidades.

Ahora bien, desde siempre han existido alternativas de bajo costo para la automatización de, al menos, el catálogo. El Microsis ha sido la punta de lanza en esta labor, a pesar de sus limitaciones. Gracias al empeño de muchos especialistas de diferentes campos, han surgido nuevas aplicaciones basadas en los principios del Software Libre. Estos permiten la automatización integral de todos los servicios y los procesos realizados en la UI y la generación de servicios agregados a las personas usuarias y al desarrollo de bibliotecas virtuales.

La escogencia y la utilización de una aplicación dependerán, en gran medida, de los objetivos e intereses de la UI, así como de sus recursos. Si bien es cierto que con el *software* libre se pueden realizar proyectos de bajo costo, esos costos no se eliminan del todo y es necesaria cierta inversión en infraestructura tecnológica y en recurso humano.

2. Software Libre

El proyecto *software* libre nace en 1983 con el GNU³ de Richard Stallman, investigador del Laboratorio de Inteligencia Artificial del

³ GNU es un acrónimo recursivo que significa GNU *is Not Unix*

Massachusetts Institute of Technology. Stallman inició este proyecto con la intención de crear un sistema operativo compatible con Unix, pero de código abierto, de forma tal que cualquier persona pudiese utilizarlo, modificarlo y adaptarlo a sus necesidades. En 1985, Stallman funda la *Free Software Foundation* (FSF) con la idea fundamental de promover y difundir el *software* de código abierto. Su meta principal consiste en “promocionar la libertad de los usuarios de ordenadores, y defender los derechos de los usuarios de software libre” (Free Software Funtation, 2010).

La FSF publica en 1989 la primera versión de GNU General Public License (GNU GLP), la cual está orientada a “garantizar la libertad de compartir y modificar todas las versiones de un programa, con el fin de asegurarse de que sigue siendo software libre para todos sus usuarios” (GNU, 2007).

Es importante aclarar que el *software* gratuito no es sinónimo de *software* libre. Existe gran variedad de *software* gratuito, el cual es posible descargar y utilizar sin ninguna restricción; sin embargo, no se tiene acceso al código fuente y por lo tanto, no puede ser modificado y adaptado.

El *software* libre se define como aquella aplicación que es adquirida sin pagar y es suministrada junto con el código fuente, el cual puede ser copiado, distribuido, modificado y adaptado a las necesidades de las personas usuarias.

Arriola (2008) es un poco más específico y da cuatro pautas que debe cumplir:

- La libertad de usar el programa con cualquier propósito (libertad 0).
- La libertad de estudiar cómo funciona el programa y adaptarlo a sus necesidades (libertad 1). El acceso al código fuente es una condición previa para esto.
- La libertad de distribuir copias, con lo que se puede ayudar a otros colegas (libertad 2).
- La libertad de mejorar el programa y publicar las mejoras de modo que toda la comunidad se beneficie (libertad 3). El acceso al código fuente es un requisito previo para esto.

Por tal motivo, tenemos un panorama muy amplio en la descarga y utilización del *software* y su adaptación a nuestras necesidades de automatización.

3. Aplicaciones de Software Libre para Unidades de Información

La automatización de bibliotecas nació en los sesenta, en las bibliotecas universitarias de los Estados Unidos; particularmente, con sistemas de procesamiento por lotes (*batch*). Pero el impulso más grande lo recibió de la Biblioteca del Congreso de los Estados Unidos, la cual creó, a principios de los setenta, el formato MARC (Machine Readable Cataloging),

un formato de registro con todos los campos requeridos para contener la información bibliográfica de todo tipo de documento.

Por su parte, a principios de los años sesenta, la Organización Internacional del Trabajo estudiaba la forma de mecanizar sus procesos bibliotecarios. Desarrolló el Integrated Set of Information System (ISIS), el cual distribuyó en el nivel internacional. Dicho software evolucionó, posteriormente, sobre los nuevos equipos con su versión MiniIsis, y, durante la década de los ochenta y de la mano de la UNESCO, se desarrolló el MicroIsis (o Micro CDS/ISIS) para microcomputadoras y sistema operativo MS-DOS (Mejía, 2010).

Hoy, existe una amplia gama de programas para uso en las bibliotecas basados en *software* libre. El idóneo dependerá de los objetivos de la implementación buscados. Por ejemplo, existen desde programas sencillos para la automatización del catálogo público hasta sistemas complejos de automatización integral de procesos, servicios y productos.

Se tienen tres grandes áreas bien diferenciadas en la automatización de Unidades de Información y sobre las que se agrupan los diferentes programas de aplicación: la automatización de catálogos, las bibliotecas virtuales y los repositorios electrónicos, y los sistemas integrados de automatización de bibliotecas.

3.1. Sistemas de Automatización de Catálogos

Son programas que se concentran en la automatización, únicamente, del módulo de catalogación. Son, hasta cierto punto, programas pequeños y versátiles que utilizan un manejador de base de datos no relacional y tablas planas. Su estructura simple permite una enorme rapidez en la realización de consultas por medio de archivos invertidos con punteros a los datos y sobre los que se corren las consultas, procesando cantidades ingestas de registros en pocos segundos. Sin embargo, al ser sistemas no relacionales, no es posible tener restricciones de integridad en los registros, por lo que no es posible la relación entre tablas.

Al usar este tipo de *software*, siempre es importante la utilización, en el diseño de los metadatos, de formatos de intercambio como MARC, con el fin de poder migrar a plataformas más robustas de forma más sencilla y ordenada. El uso de formatos de intercambio de información permite la comunicación y migración entre diferentes plataformas de forma transparente, al existir una base de intercambio para recibir los datos.

Los programas de mayor difusión en este tipo de automatización son el WinIsis, con una larga tradición en la automatización de bibliotecas, sus versiones anteriores de MiniIsis (para sistemas cerrados de minicomputadoras) y MicroIsis (para MS-DOS) y ahora, el Isis-Marc.

El WinIsis, desde su base de procesamiento, se mantiene muy activo y se han desprendido de él otros programas de automatización que lo utilizan como núcleo y que se mencionarán más adelante.

3.1.1. WinIsis

Básicamente, el WinIsis se compone de una sencilla interfaz gráfica y una serie de archivos sobre los cuales se diseñan y administran las bases de datos.

Los archivos de administración son cuatro: la tabla de definición de campos (FDT), donde se especifican los metadatos; la hoja de entrada de datos (FMT), que permite definir las vistas para el ingreso de datos; el formato de visualización (PFT), que permite diseñar la forma en que se desplegarán los registros tras una consulta; y la tabla de selección de campos (FST), donde se especifican los puntos de acceso que formarán parte del archivo invertido.

Por su parte, cada base de datos se compone de tres archivos: el archivo maestro, el cual contiene todos los registros identificados por un número único asignado automáticamente por el sistema (Master File Number, MFN); el archivo de referencias cruzadas, el cual es un índice donde se especifica cada registro dentro del archivo maestro; y el archivo invertido (también denominado Diccionario de datos e índice de contenidos), el cual consiste en una serie de puntos de acceso o punteros definidos por la persona usuaria en la FST (autor, título, descriptores, signatura, etc.), y una lista de referencias hacia el archivo maestro.

A pesar de ser un *software* antiguo, su sistema de recuperación por punteros es sumamente eficiente y se utiliza ya no solo como software de automatización de catálogos, sino como base de búsquedas en la Internet para *DataWharehousing*.

3.1.2. Isis-Marc

El Isis-Marc, con licencia GNU-GLP, es una interfaz diseñada, específicamente, para la catalogación bajo el formato MARC 21. Es un utilitario de WinIsis (tipo *add-on*) que reemplaza la ventana de entrada de datos. Tiene la ventaja de que posee una estructura de campos y subcampos predefinida y configurable, basada en MARC21 (FDT21). Trae incorporada la base de datos CODES, la cual implementa los códigos MARC, así como una base de datos de autoridad con los parámetros de estandarización para el ingreso de los autores y las autoras. Esta última característica es de suma importancia, ya que permite la manipulación de archivos de autoridad como bases de datos externas.

Otras características interesantes son:

- La catalogación por copia.
- La inclusión del cliente Z39.50, por lo que es posible descargar la descripción bibliográfica completa de servidores externos.
- El manejo de perfiles de personas usuarias.

Figura No. 1
Ventana principal de IsisMarc

Etq	Campo	Contenido
001	Número de control	00089048230 /AC/r91
003	Identificador del número de control	DLC
005	Fecha de la última modificación	19911106082810.9
008	Campo de control - Libros	891101s1990 maua j 001 0eng
010	LoC - Número de control	##^a##89048230 /AC/r91
015	Número de la bibliografía nacional	##
017	Número de depósito legal	##
020	International Standard Book Number	##^a0316107514
020	International Standard Book Number	##^a0316107506 (pbk.)
022	International Standard Serial Number	##
024	Otro identificador normalizado	##
028	Número del editor	##
035	Número de control del sistema	##
037	Fuente de la adquisición	##
040	Origen de la catalogación	##^aDLC^cDLC^dDLC
041	Código de idioma	##
043	Código de área Geográfica	##
050	Número de Clasificación de la Biblioteca	00^aGV943.25^bB74 1990
060	Número de Clasificación de la Biblioteca	##
066	Conjunto de Caracteres Presente	##
072	Código de categoría temática	##
080	Número de Clasificación Decim	##

Fuente: Software IsisMarc

3.2. Aplicaciones para Bibliotecas Virtuales y Repositorios Electrónicos

Este tipo de aplicaciones se caracterizan, primero, por estar orientadas al almacenamiento y recuperación de documentos electrónicos en repositorios y, segundo, por proporcionar una serie de servicios propios de una biblioteca física. No son programas orientados a la catalogación propiamente dicha o a la automatización integral de bibliotecas. Su función es la de proporcionar una interfaz amigable para la recuperación de documentos agrupados de acuerdo con normas e intereses definidos por la persona usuaria.

3.2.1. iAH

La interfaz del iAH (Interface for Access on Health Information) es un programa multiplataforma basado en MicroIsis y diseñado para recuperar, vía Internet, información de las bases de datos diseñadas sobre dicha aplicación. Ha sido desarrollada por BIREME (Biblioteca Regional de Medicina), y a partir de Isis Script.

La recuperación en la base de datos puede ser realizada a través de expresiones booleanas y acceso a los índices de los archivos invertidos.

La Interfaz del iAH está estructurada en tres partes, las cuales son:

- Área de Documentos (DOCUMENT_ROOT): localización primaria de los documentos (HTML, java script, CSS, imágenes, archivos de configuración) de la aplicación en el servidor *web*.
- Área de Ejecución de Scripts (CGI-BIN): esta área se encarga de la localización de los archivos ejecutables, programas o *scripts* de la aplicación. Aquí también se localiza el código de la aplicación escrita en lenguaje Isis Script (compilado o no) y el del código WWWISIS ejecutable.
- Área de bases de datos: esta área localiza las bases de datos que serán utilizadas por la aplicación.

Figura No. 2
iAH: formulario básico de consulta aplicado en la Biblioteca Virtual en Población y Salud



B V P

Búsqueda en bases de datos

ayuda Base de datos: TEXTO Formulario libre

Buscar por : [Formulario básico](#) [Formulario avanzado](#)

Ingrese una o más palabras

☒ Todas las palabras (AND) ☐ Cualquier palabra (OR)

config búsqueda

Fuente: Biblioteca Virtual en Población y Salud <http://ccp.ucr.ac.cr/bvp>

Figura No.3
iAH: formulario avanzado de consulta aplicado en la Biblioteca Virtual en Población y Salud

The screenshot displays the 'Formulario avanzado' (Advanced Form) for searching in the 'Base de datos : TEXTO'. The interface includes a search bar labeled 'Buscar' and a dropdown menu for 'en el campo' currently set to 'Palabras'. Below this, there are three rows for constructing a query, each starting with 'and' and followed by a search field. To the right of each search field is a small icon and the word 'índice'. At the bottom of the form are three buttons: 'config', 'limpia', and 'búsqueda'.

Fuente: Biblioteca Virtual en Población y Salud <http://ccp.ucr.ac.cr/bvp>

3.2.2. Greenstone

Greenstone es un *software* multiplataforma elaborado en 2000 como parte del proyecto de la Biblioteca Digital de la Universidad de Waikato, Nueva Zelanda. Actualmente, es un *software* de código abierto bajo los términos y condiciones de la Licencia Pública General de GNU, desarrollado y distribuido en colaboración con la UNESCO y la ONG Human Info.

Greenstone se compone de un conjunto de colecciones digitales, las cuales son construidas importando documentos de una gran variedad de formatos (MS-Office, PDF, HTML, correo electrónico, CDS-ISIS, OpenOffice, etc.). Toda la información textual importada es procesada y transformada en ficheros de formato XML con caracteres Unicode, UTF-8.

El proceso de construcción de las colecciones se compone de cinco fases:

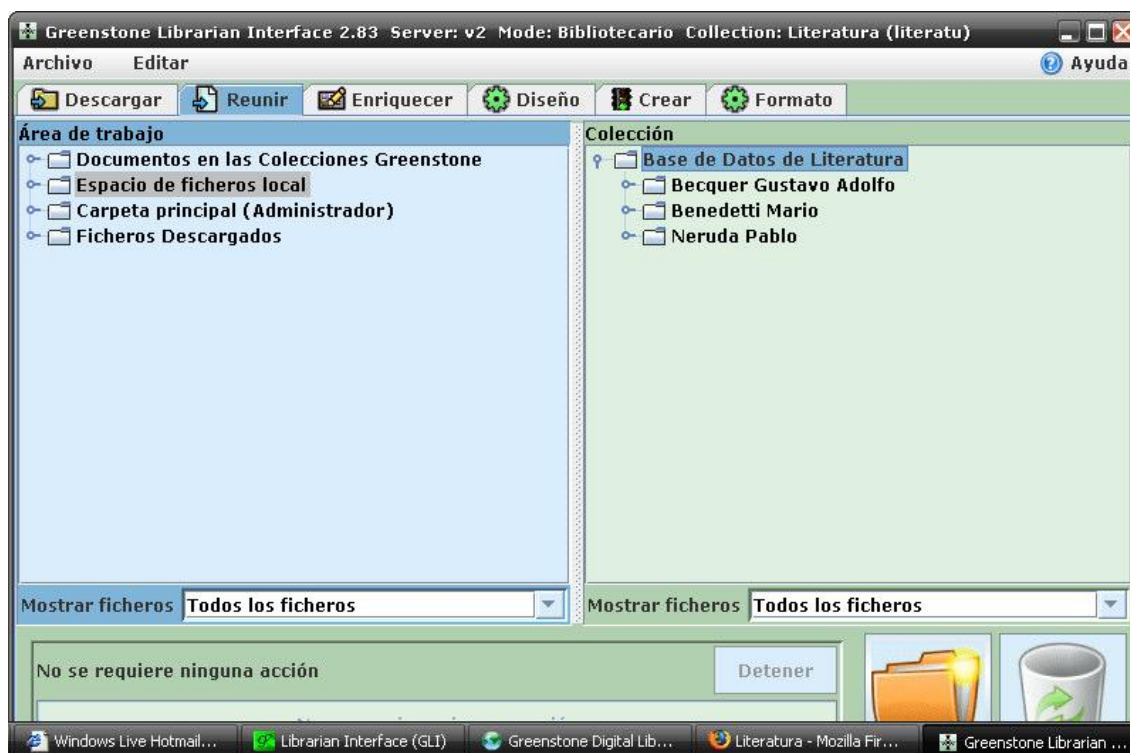
1. Creación: se define el nombre, se describe su contenido y responsable de la gestión.
2. Selección de documentos: especifica los documentos que se van a incluir en la colección y su ubicación física dentro del servidor de archivos.
3. Configuración de la colección: parámetros básicos de tratamiento, indización y presentación de la colección.
4. Construcción de la colección: los documentos son importados, procesados por los *plugins* correspondientes, transformados en xmls

e indizados. Crea índices textuales y estructuras de organización y acceso (llamadas *clasificadores*).

5. Consulta a la colección: es un proceso de localización y acceso de los documentos que utiliza la interfaz disponible, por medio de búsquedas en los índices textuales o en los clasificadores.

Este software permite la administración de colecciones electrónicas para ser accedidas de forma rápida y eficiente.

Figura No.4
Greenstone: interface para el profesional bibliotecario



Fuente: Greenstone Digital Library Software

3.2.3. Google Custom Search

La corporación Google ha incursionado fuertemente en el área de la digitalización y la construcción de bibliotecas virtuales. Ha desarrollado diferentes aplicaciones para la búsqueda, la recuperación y el despliegue de documentos electrónicos; esta labor marca un hito en la evolución y la difusión de este tipo particular de formato de información.

El Google Custom Search es una de las aplicaciones gratuitas que, si bien está concebida como apoyo a la indexación y búsqueda interna dentro de portales, puede utilizarse como apoyo a la biblioteca virtual. Se trata de un buscador personalizado que se puede colocar en cualquier tipo de sitios *web* y permite indexar hasta 5000 páginas o cualquier tipo de documento

ubicado en una dirección determinada. Esta última opción es sumamente interesante, ya que si tenemos un conjunto de documentos ubicados en una carpeta pública, la aplicación los indexará y proporcionará una interfaz de búsqueda remota, sin necesidad de instalar de forma local programa alguno.

Figura No.5

Aplicación de Google Custom Search en la revista Población y Salud en Mesoamérica



Fuente: Población y Salud en Mesoamérica
<http://ccp.ucr.ac.cr/revista/busqueda.htm>

3.2.4. DSpace

Este software fue desarrollado por el MIT (Massachusetts Institute of Technology) y la corporación Hewlett Packard, se utilizó PostgreSQL como motor de base de datos. Inicialmente, se concibió como un repositorio para que los investigadores del MIT “colgaran” y compartieran los archivos de sus investigaciones. Dichos archivos pueden ser de cualquier tipo (como texto, audio, vídeo y presentaciones) y son organizados mediante metadatos basados en el formato de intercambio Dublin Core.

Este programa tiene la particularidad de crear y administrar comunidades y sub-comunidades de acuerdo con la temática que se desee investigar.

Figura No.6
Interface de consulta de DSpace: repositorio del
Instituto Tecnológico de Costa Rica



Fuente: Instituto Tecnológico de Costa Rica
<http://bibliodigital.itcr.ac.cr:8080/dspace/>

3.3. Sistemas Integrados de Automatización de Bibliotecas

Estos sistemas están dirigidos a la gestión de cada una de las áreas que componen una biblioteca. Son sistemas complejos, concebidos bajo una arquitectura modular y utilizan bases de datos relacionales.

Para que un software pueda ser considerado parte de esta categoría, debe utilizar motores de bases de datos relacionales y ofrecer soluciones a las grandes áreas que componen una biblioteca: adquisiciones, catalogación (que incluye los catálogos), circulación, publicaciones periódicas, consulta e inventario. Es recomendable, además, un módulo de estadísticas, sobre las cuales tomar decisiones importantes para el manejo de la biblioteca, por ejemplo, la dirección del presupuesto a áreas de mayor utilización de recursos bibliográficos, horarios de mayor volumen de transacciones, etc.

Actualmente, la mayoría de los programas de acceso abierto para la automatización integral se basan en la tecnología *Web*, son programados en lenguajes tipo GNU, se fundamentan en el formato MARC21, utilizan motores de bases de datos gratuitos y potencian las capacidades de gestión y recuperación remota. Inicialmente, han sido programados para

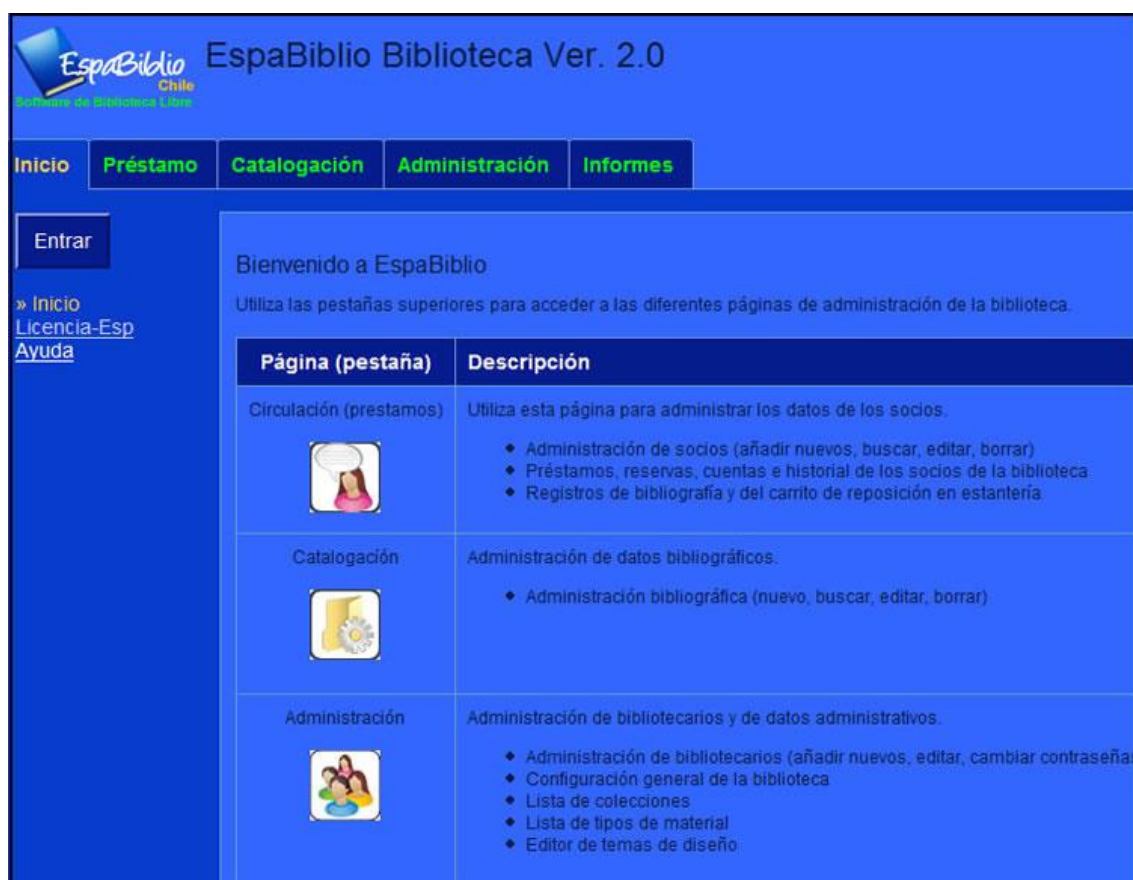
“correr” sobre Linux como sistema operativo base; sin embargo, muchos cuentan con versiones multiplataforma.

Al ser sistemas basados en *Web*, tienen requerimientos de programas adicionales para funcionar, como servidores *web* (Apache, IIS, TomCat, etc), motores de bases de datos (MySQL, PostgreSQL, Jet, etc), módulos Java, entre otros.

3.3.1. EspaBiblio

Este software es la versión en español del OpenBiblio. Utiliza MySQL como motor relacional de base de datos y se encuentra totalmente desarrollado en el lenguaje PHP. Es un software sumamente amigable e intuitivo y posee los módulos de circulación, procesos técnicos, administración e informes.

Figura No.7
Pantalla de inicio de EspaBiblio



Fuente: EspaBiblio. Sistema de Gestión y Automatización de Bibliotecas
<http://www.desem.cl/espabiblio/imagenes.php>

3.3.2. Koha

Koha es una de las soluciones más completas existentes en el mercado. Se basa en tecnología de intranet, por lo que es posible separar la administración general de la biblioteca del catálogo público en línea; en consecuencia ofrece un importante parámetro de seguridad. Está desarrollado mediante lenguaje Perl y utiliza MySql como motor de base de datos.

Figura No.8
Interface de consulta de Koha: Universidad Nacional de la Plata



Fuente: Grupo de desarrollo de Koha-UNLP

3.3.3. Open MarcoPolo

Open MarcoPolo es una solución interesante para aquellas unidades de información que basan sus catálogos en WinISIS. Está desarrollado mediante WXIS (WWWISIS XML ISIS Script Server), HTML y JavaScript. Utiliza el motor de base de datos de WinISIS, el cual no es relacional, por lo que debe generar diferentes bases de datos independientes para su funcionamiento y administración.

Figura No.9
Interface principal de OpenMarcoPolo



Fuente: Software Open Marco Polo

4. Ventajas y desventajas

La automatización de catálogos y bibliotecas y la creación de bibliotecas virtuales plantean retos; es necesario tener muy claros el objetivo que se persigue, los requerimientos necesarios y las necesidades reales de la comunidad usuaria que se pretende servir. También resulta indispensable contar con un presupuesto, ya que aunque el *software* sea gratuito, se requiere invertir en equipo y recurso humano. De acuerdo con los objetivos, es posible determinar qué tipo de *software* se debe emplear, ya sea para automatizar un catálogo o todos los módulos de la biblioteca.

4.1. Ventajas

Indudablemente, la principal ventaja obtenida con la utilización del *software* libre la constituye el ahorro en la compra de licencias. Ese tipo de programas es sumamente costoso y demanda de grandes presupuestos para su implementación y funcionamiento. Generalmente, solo las bibliotecas con fuertes fuentes de financiamiento, como las universitarias, poseen la capacidad para adquirir este tipo de soluciones.

Otra gran ventaja se halla en el hecho de que existen soluciones para cada nivel de automatización en el que se desee involucrar a la organización (automatización de catálogos, bibliotecas virtuales o automatización integral). Asimismo, se dispone de soluciones sencillas hasta complejos sistemas de gestión.

Como respuesta a la falta de un soporte técnico incluido en la compra de un *software*, los programas tipo GNU cuentan con una enorme comunidad usuaria que desarrollan e intercambian experiencias, comunican sus problemas y llegan a soluciones conjuntas. Las notificaciones se dan por medio de listas de correos, foros de discusión, *blogs*, redes sociales y portales específicos; por lo que la oferta de ayuda, si bien no siempre es inmediata, es muy amplia.

Muchas veces, instalar y poner en marcha un proyecto de este tipo hace que las personas profesionales en Bibliotecología se identifiquen con el *software* y terminen convirtiéndose en verdaderas expertas, por lo que proponen mejoras, realizan adaptaciones y promueven su utilización.

4.2. Desventajas

La principal desventaja al utilizar *software* libre la constituye el no contar con una casa comercial que brinde una guía y el soporte técnico al proyecto. Por lo general, al comprar una solución, se incluye, dentro del contrato, la capacitación y el soporte técnico durante un periodo determinado, lo cual permite familiarizarse con la aplicación y disminuir el tiempo invertido en el aprendizaje. Con ello, es posible tener el *software* en producción en un periodo relativamente corto.

La instalación y puesta en marcha de un programa basado en *software* libre requiere de mucho esfuerzo y aprendizaje. La dedicación a esta labor debe ser grande y ardua y requerirá de mucho tiempo. Por otro lado, si se desea contratar personal para tal fin, el desarrollo de la aplicación resulta sumamente costoso debido a que los especialistas en dicho *software* son escasos y, por lo tanto, sus conocimientos, muy cotizados.

La implementación de una aplicación de este tipo, generalmente, viene acompañada de una serie de requerimientos de *software* que muchas veces son más difíciles de instalar y administrar que el programa de automatización en sí. Deberán instalarse los motores de base de datos, los programas de desarrollo (por ejemplo, Java, Perl, etc.) y los complementos específicos para el correcto funcionamiento. Por tal motivo, se recomienda que sean “corridos” sobre Linux para facilitar la instalación y aprovechar todas sus potencialidades.

Por lo general, el personal bibliotecario no cuenta con una sólida formación en el área informática, por lo que deberá enfrentarse a la disyuntiva de iniciar el aprendizaje, en solitario, del sistema operativo y el software de aplicación o dejar en manos del personal de informática la implementación. Este segundo tipo de personal tiene un rol de trabajo arduo en el área de soporte y pocas veces está dispuesto a realizar labores que impliquen destinar tiempo a la investigación.

Por lo general, el tiempo destinado para implementar una solución de este tipo siempre será mucho más largo, debido a la necesidad inicial de

conocer y adaptar la aplicación, acondicionar el *hardware*, estudiar los manuales de usuario (si los tiene), contactar a las comunidades usuarias y familiarizarse plenamente con el programa. Por su parte, una solución comprada se realiza bajo la modalidad de llave en mano, donde la persona que vende instala, realiza las pruebas, acondiciona el programa y lo entrega en plena producción.

5. Conclusiones

Cualquier proyecto de automatización siempre será un proceso que representa mucho trabajo, pero lleno de retos y satisfacciones. Si se desea iniciar un proyecto de esta índole, será necesario seguir el proceso de recopilación de requerimientos, análisis y diseño lógico de la alternativa, tomando muy en cuenta el presupuesto que se tiene y el tiempo que se requiere para finalizar la meta.

Es importante recalcar que no existe una solución gratuita. Si bien es cierto la compra de una solución puede resultar sumamente costosa, los programas tipo GNU, aunque no sea necesario pagar por su utilización, implican una serie de costos provenientes de la renovación de equipo, capacitación en el uso del *software*, conversiones, investigación y posterior operación y producción.

Aún así, las alternativas de *software* libre son múltiples, eficientes y cubren todos los niveles a los que se desee llegar, desde un simple catálogo automatizado hasta una compleja aplicación de gestión bibliotecaria. Por tal motivo, este tipo de soluciones deben ser tomadas en cuenta cuando se pretende iniciar un proyecto.

Lo más recomendable al instalar este tipo de software es utilizar su sistema operativo nativo; esto es, si inicialmente ha sido diseñado para Linux, es recomendable que sea instalado sobre dicha plataforma. A pesar de que los programas se diseñan para ser multiplataforma, las versiones para otros sistemas operativos, por lo general, van retrasadas en cuanto a mejoras y nuevas aplicaciones. Por otro lado, el *software* requiere, para funcionar, de muchos complementos que el sistema operativo nativo ya posee, por lo que implicará más trabajo de instalación, configuración y puesta a punto de la aplicación.

6. Referencias bibliográficas

Arriola, Ó.; Butrón, K. (2008). Sistemas integrales para la automatización de bibliotecas basados en software libre. *ACIMED* 18(6). Consulta del 30 de junio 2010 de http://www.bvs.sld.cu/revistas/aci/vol18_6_08/

BIREME (2006). *Manual del usuario de la IAH*. Sao Pablo: BIREME / OPS / OMS

Espabiblio: Sistema de Gestión de Bibliotecas. (s.f.). Consulta del 11 de agosto, 2010 de <http://www.desem.cl/espabiblio/>

Free Software Foundation. (s.f.). Consulta del 26 de junio, 2010 de <http://www.fsf.org/>

GNU General Public License. (s.f.). Consulta de 26 de junio de 2010 de <http://www.gnu.org/copyleft/gpl.html>

Grupo de desarrollo de Koha-UNLP (2009). *Koha: sistema integrado de gestión de bibliotecas.* Argentina: Universidad Nacional de La Plata

Lee, M. (2010). What is free software and why is it so important for society?. *Free Software Foundation.* Consulta del 26 de junio, 2010 de <http://www.fsf.org/about/what-is-free-software>

Mejía, J. (2010). Sistemas integrales para la automatización de bibliotecas basados en software libre. *Exágono bibliotecario.* Consulta del 30 de junio, 2010 de <http://exagonobibliotecario.blogspot.com/2010/03/sistemas-integrales-para-la.html>

Portal del proyecto Open Marco Polo. (s.f.). Consulta del 11 de agosto, 2010 de <http://marcopolo.uner.edu.ar/>

Rodríguez, J.; Sulé, A. (2008). DSpace: un manual específico para gestores de la información y la documentación. *Bid* (20), junio

Saorín, T. (2002). *Modelo conceptual para la automatización de bibliotecas en el contexto digital.* Murcia: Universidad de Murcia.